The Multi-Period Service Territory Design Problem - An Introduction, a Model and a Heuristic Approach

Citation for published version:
Bender, M, Meyer, A, Kalcsics, J & Nickel, S 2016, 'The Multi-Period Service Territory Design Problem - An Introduction, a Model and a Heuristic Approach', *Transportation Research Part E: Logistics and Transportation Review*, vol. 96, pp. 135-157. https://doi.org/10.1016/j.tre.2016.09.007

Digital Object Identifier (DOI):
10.1016/j.tre.2016.09.007

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Transportation Research Part E: Logistics and Transportation Review

The Multi-Period Service Territory Design Problem – An Introduction, a Model and a Heuristic Approach

Matthias Bender[a,*], Anne Meyer[a], Jörg Kalcsics[b], Stefan Nickel[a,c]

[a] *Department of Logistics and Supply Chain Optimization, Research Center for Information Technology (FZI), Karlsruhe, Germany*
[b] *School of Mathematics, University of Edinburgh, Edinburgh, Scotland*
[c] *Institute of Operations Research, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany*

Abstract

In service territory design applications, a field service workforce is responsible for providing recurring services at their customers' sites. We introduce the associated planning problem, which consists of two subproblems: In the partitioning subproblem, customers must be grouped into service territories. In the scheduling subproblem, customer visits must be scheduled throughout the multi-period planning horizon. The emphasis of this paper is put on the scheduling subproblem. We propose a mixed integer programming model for this subproblem and present a location-allocation heuristic. The results of extensive experiments on real-world instances show that the proposed heuristic produces high-quality solutions.

Keywords: territory design, multi-period planning horizon, mixed integer linear programming, location-allocation heuristic

2010 MSC: 90B06, 90B80, 90C59

1. Introduction

Many companies employ a field service workforce for providing recurring services at their customers' sites. For example, manufacturers and wholesalers of consumer goods typically operate a sales force that regularly visits their customers to promote sales or to supply product range information (see, e.g., Fleischmann and Paraschis, 1988; Polacek et al., 2007). Also, some engineering companies employ field service technicians to carry out regular technical maintenance at their customers' sites (see, e.g., Blakeley et al., 2003). The frequency and duration of the visits depend on customer-specific factors, e.g., the customer's sales volume or the tasks to be performed at the customer. To increase customer satisfaction, two aspects of service consistency play an important role in these applications: personal and temporal consistency. The former means that always the same field worker is responsible for a particular customer, which is desirable as it helps establish and foster long-term personal relationships with customers (see, e.g., Kalcsics et al., 2005; López-Pérez and Ríos-Mercado, 2013; Zoltners and Sinha, 2005). The latter expresses the expectation of

*Corresponding author

Email addresses: mbender@fzi.de (Matthias Bender), meyer@fzi.de (Anne Meyer), joerg.kalcsics@ed.ac.uk (Jörg Kalcsics), stefan.nickel@kit.edu (Stefan Nickel)

customers to be visited on a regular basis (see, e.g., Groër et al., 2009, for a similar consistency requirement arising in the small package shipping industry). Regularity means, on the one hand, that the visits should be equally distributed over the weeks of the planning horizon according to customer-specific visiting rhythms. On the other hand, regularity refers to the weekdays on which visits take place as customers might prefer to be served always on the same weekdays.

Typically, the following three planning tasks arise in these applications. (1) The customer base must be partitioned into service territories with one field worker being responsible for each territory. This partition is usually maintained over a long period of time to promote the development of personal relationships between field workers and customers. (2) On a tactical level, the visit schedules have to be created, which means that the visiting days for each customer must be determined. The planning horizon for this task is typically between 3 and 12 months. (3) On an operational level, the detailed planning must be performed, which includes the planning of the daily routes and, when necessary, the rescheduling of visits. It is important to note that short-term customer requests and unexpected events must be taken into account in this step. According to estimates of our project partner, about 20% of the customer visits need to be rescheduled to another day in the short term. Therefore, both the route planning and the rescheduling are done by the field worker in the daily business. Ideally, planning tasks (1) to (3) would be tackled by a single, integrated approach, but the size of realistic problem instances (sometimes with ten thousand or more customers) prohibits an integrated approach. Moreover, integrating the calculation of the daily routes and the visit schedules is only of little use due to the potential necessity to reschedule customer visits in the daily business.

The above problem was brought to our attention by our project partner PTV Group, a commercial provider of districting and clustering software headquartered in Karlsruhe, Germany. In our joint project, we tackled the partitioning task (1) and the scheduling task (2); we omitted the routing and rescheduling task (3) as this task can only be solved reasonably in the short term when all operational details are known.

One of PTV's products is the xCluster Server (PTV, 2014), which solves the optimization problem resulting from the scheduling task (2). When the planning algorithm for the xCluster Server was initially designed several years ago, the technological possibilities were limited, in particular with regard to the availability of high-performance mixed integer programming (MIP) solvers and computational power in general, which lead PTV to develop a simple local search procedure. The goal of the cooperation with PTV is the development of a new solution approach that takes advantage of recently available technologies. Since PTV has many different customers, it is important that the new solution approach covers a wide range of real-world requirements. Additionally, it must be easily adaptable to further planning requirements. The new approach is intended to replace the existing planning algorithm in the xCluster Server.

The main contributions of this paper are the following:

- We introduce a new problem, which we call the Multi-Period Service Territory Design Problem (MP-STDP). Despite its high practical relevance, it has not been studied in the literature before. This is, to the best of our knowledge, the first paper to elaborate the problem from a scientific point of view.

- We formally define the scheduling subproblem, i.e., the subproblem corresponding to planning task (2), as a mixed integer linear programming model.
- We propose a heuristic solution approach for the scheduling subproblem. The approach is capable of considering the relevant planning requirements of PTV's customers. It involves the repeated solution of an integer programming model, which can easily be extended by additional planning requirements.
- We perform extensive computational experiments on real-world instances and on instances that were derived from real-world data by varying the values of some parameters. The results show that the new approach produces high-quality solutions and outperforms the existing solution method of PTV.

The remainder of this paper is organized as follows. In Section 2 we give a detailed description of the problem under study. In Section 3 we review related problems and point out the differences to our problem. In the subsequent section, we introduce a mathematical model for the subproblem that corresponds to the scheduling task (2). In Section 5 we propose a heuristic approach based on a location-allocation scheme. To evaluate our approach, we introduce appropriate evaluation measures in Section 6. In Section 7 we report the results of extensive computational experiments on real-world data and benchmark our approach against PTV's xCluster Server (PTV, 2014). Finally, we provide some concluding remarks in Section 8.

2. Problem Description

In this section, we describe the MPSTDP and introduce the notation for the scheduling subproblem, which is the major focus of this paper.

There is a given set of *customers* (e.g., supermarkets), represented by index set $B = \{1, ..., |B|\}$, which demand recurring on-site services. The services must be carried out by a given set of field workers, which we call *service providers*. Corresponding to planning tasks (1) and (2), the MPSTDP consists of the following two subproblems.

Partitioning subproblem (MPSTDP-P): This subproblem corresponds to the well-known territory design or districting problem (see Kalcsics, 2015, for an overview of typical planning criteria). The set of customers must be partitioned into service territories with exactly one service provider being responsible for each service territory. As the service providers have to travel within their territories, geographically compact and connected territories are desired because they lead to short travel times for the service providers. Furthermore, for reasons of fairness, all service territories should have approximately the same workload.

Scheduling subproblem (MPSTDP-S): In this subproblem, a valid visit schedule must be determined for each service territory, i.e., customer visits must be assigned to the weeks and days of the planning horizon subject to customer-specific visiting requirements. The planning horizon comprises $|W|$ weeks and m days per week, resulting in $m|W|$ days in total. Weeks and days are indexed by $w \in W = \{1, ..., |W|\}$ and $d \in D = \{1, ..., |D|\}$, respectively. The customer-specific visiting requirements restrict the temporal distribution of customer visits at two levels.

At the level of weeks, the visits of each customer must be periodically recurring according to a customer-specific *week rhythm* $r_b \in \mathbb{N}^+$, $b \in B$, meaning that each customer $b \in B$ must be visited every r_b weeks. We call a week in which a customer is visited by a service provider a *visiting week* of the customer. As the first visit of each customer $b \in B$ must be in the first r_b weeks of the planning horizon, a customer's week rhythm can be translated into r_b valid combinations of visiting weeks P_b, which we call *week patterns*. If, for example, a customer's week rhythm is $r_b = 2$ and the planning horizon consists of $|W| = 6$ weeks, P_b contains the week patterns $\{1, 3, 5\}$ and $\{2, 4, 6\}$, i.e., the customer must be visited either in weeks one, three and five or in weeks two, four and six.

At the level of days, there are restrictions on the number of visits per visiting week and on the weekdays on which customers may be visited. More precisely, each customer $b \in B$ must be visited n_b times in each visiting week. A day on which a customer is visited is said to be a *visiting day* of the customer. The visiting days within each visiting week must correspond to one of the customer's valid *weekday patterns* Q_b. A weekday pattern is a combination of weekdays on which the customer may be visited. For example, for a customer with $n_b = 2$, the set Q_b could consist of the weekday patterns {Monday, Thursday} and {Tuesday, Friday}, meaning that the customer must be visited either on Monday and Thursday or on Tuesday and Friday. Additionally, if regularity is required with respect to the weekdays on which a customer is visited, we call this a *weekday regularity* of the customer.

The number of weeks in the planning horizon, $|W|$, is typically chosen as the least common multiple of the week rhythms r_b, $b \in B$ since, after this time, the schedule could be repeated identically. Therefore, a customer $b \in B$ must be visited $\frac{|W|}{r_b} n_b$ times during the entire planning horizon. Each visit of a customer requires an individual service time. By t_{bj}, $j \in \{1, ..., \frac{|W|}{r_b} n_b\}$ the service time associated with the j-th visit of customer $b \in B$ is given.

When customer visits are scheduled, compactness – in the sense of geographically concentrated customer visits – plays a crucial role. As in the partitioning subproblem, this is again due to the fact that the service providers have to travel to their customers. On each day in the planning horizon, a service provider has to visit those customers within his or her service territory that are scheduled for that day. Hence, in order to reduce travel time, all customers that need to be visited on the same day should form a geographically compact area. Note that compactness does, of course, not necessarily lead to the shortest possible routes. In fact, there might be less compact solutions that lead to shorter travel times than a highly compact solution. But compact solutions have a significant advantage when it comes to short-term customer requests and unexpected events in the daily business as they provide a high degree of flexibility with respect to the sequence in which customers can be visited. This is illustrated by the example in Figure 1. The figure depicts the visits that are scheduled for a specific day. The right-hand side shows a fairly compact solution, whereas the solution on the left-hand side is less compact. In the example on the left-hand side, the service provider starts his route from the depot and intends to visit customer A as the first customer of the route, followed by customers B, C and D. But suppose that in the morning of that day, customer A calls the service provider

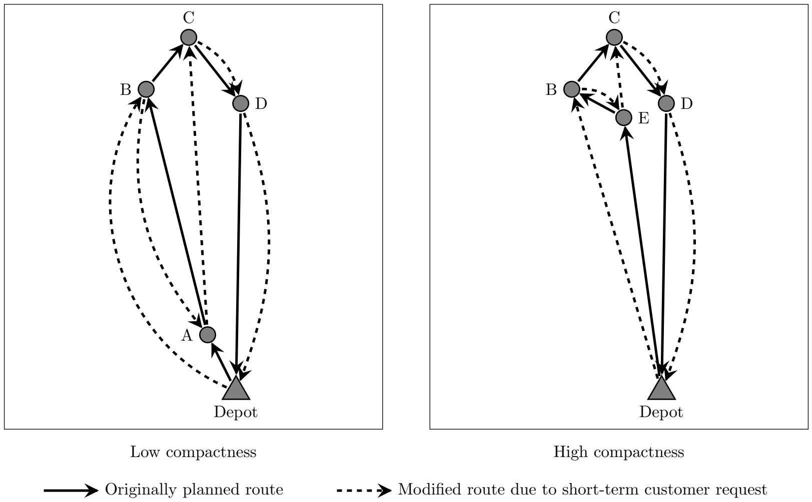


Figure 1: Compact solutions provide high flexibility with respect to the sequence in which customers can be visited, which is a highly beneficial feature when faced with short-term customer requests or other unexpected events in the daily business.

and tells him that the only possible visiting time is 12 p.m., which is in the middle of the service provider's working day. In this case, the service provider would have to visit customer B first, then travel all the way back to customer A, then visit customers C and D, and finally return to the depot. This would lead to a significant increase in travel time compared to the originally planned route and possibly even to the violation of maximum working hours. In contrast, a more compact solution, such as the example on the right-hand side of the figure, allows the service provider to fulfill short-term customer requests without a substantial increase in travel time. Suppose, for instance, that the service provider originally planned to visit the customers in the sequence E, B, C and D, and that, again, a customer visit has to be rescheduled in the short term. Let us assume in this example that customer E requests to be visited at noon, i.e., customer E cannot be visited as the first customer of the route as it was originally planned. In this case, only a small detour compared to the original plan would be necessary.

Besides the planning criterion that each service provider's daily customer visits should be geographically close to each other, there is an additional compactness requirement related to the customer visits of each week. More precisely, all customers that must be visited by the same service provider in the same week should be geographically concentrated. This requirement is motivated by the fact that, in practice, a visit which is scheduled for a certain day may not be carried out on that day, e.g., because the service provider does not arrive at the customer on time due to a traffic jam. If the customers that are scheduled for this week are geographically close to each other, the service provider can catch up on the missed visit on another day of the week without having to travel overly long distances.

The achievable compactness of the week clusters depends not only on the geographical distribution of the customers but, too a large extent, also on their week rhythms. This is illustrated by the examples in Figures 2 and 3. Let us assume for these examples that the planning horizon consists of $|W| = 2$ weeks and $m = 5$

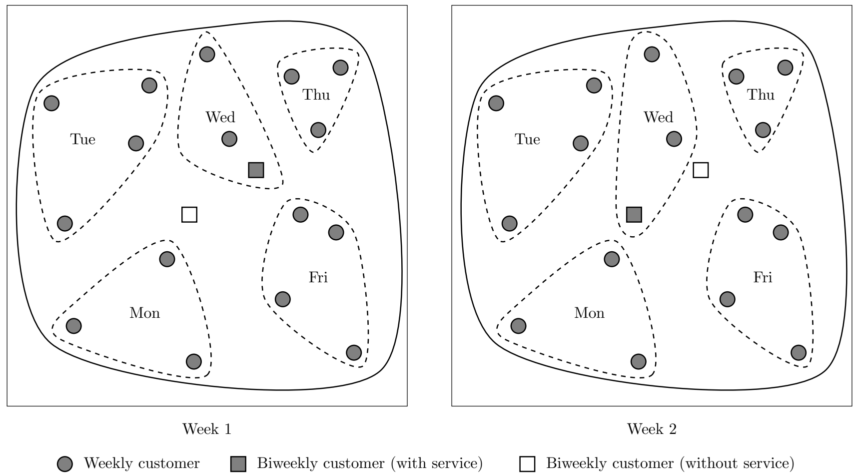


Figure 2: Solution to a problem with many weekly customers

days per week and that all customers must be visited once per visiting week, i.e., $n_b = 1$ for all $b \in B$. Figure 2 depicts the solution to a problem with almost only weekly customers that are spread evenly over the entire service territory. In this case, there exists no feasible schedule that would prevent the service provider from traveling almost all over the whole service region every week. However, when the customers' week rhythms are more favorable, it is possible to schedule the visits in such a way that the service provider needs to travel only through a relatively small area of the service territory every week. This situation is depicted in Figure 3.

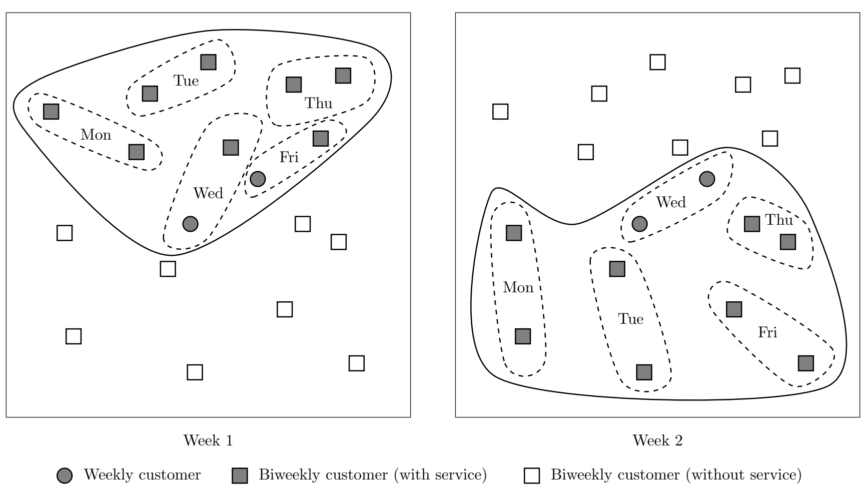


Figure 3: Solution to a problem with only few weekly customers

In order to avoid time periods with workload peaks and time periods with very little work, another important planning criterion is workload balance over time. Each service provider's workload should be evenly distributed over the planning horizon, i.e., the workload should be roughly the same on all days and in all weeks of the planning horizon.

In summary, the MPSTDP-S consists of finding a visit schedule for each service territory that satisfies the following criteria:

- The schedule is feasible with respect to the customers' visiting requirements.
- The customers to be visited on each day form a geographically compact area, which we call *day cluster*.
- The customers to be visited in each week form a geographically compact area, which we call *week cluster*.
- The service time is distributed evenly across the days of the planning horizon.
- The service time is distributed evenly across the weeks of the planning horizon.

With the aim of establishing and maintaining long-lasting customer relationships, the design of the service territories remains fairly stable over a long period of time, typically several years. As opposed to this, visit schedules are valid only for at most 12 months and, hence, have to be redetermined more frequently. Therefore, a solution approach specifically for the scheduling subproblem MPSTDP-S is required. When the schedule expires, this approach can be used to determine a new schedule without modifying the service territories. If, from time to time, the service territories need to be redesigned, we solve the subproblems MPSTDP-P and MPSTDP-S sequentially. This means that we solve a classical districting problem in the first stage. For this purpose, any existing solution method for districting problems can be used. In the second stage, we solve the scheduling subproblem by designing the week and day clusters for each service territory independently.

The partitioning subproblem MPSTDP-P has been studied extensively in the districting literature (see, e.g., Kalcsics, 2015, for a survey of applications and solution methods). However, we believe that this is the first academic work to deal with the scheduling subproblem MPSTDP-S. Therefore, we concentrate on the MPSTDP-S in the remainder of this paper.

3. Related Work

To the best of our knowledge, there are only two papers dealing with multi-period territory design problems. Lei et al. (2015) consider a problem in which the occurrence of customers changes from period to period. They assume that the customers of each period are known in advance and that a period comprises several weeks. In each period all customers must be visited exactly once on a route which starts and ends at one of the available depots. The following decisions must be made: For each period, the customers must be partitioned into districts, and a depot must be assigned to each district. Furthermore, the customers of each district must be partitioned into subdistricts with each subdistrict representing the customers that must be visited on a particular working day. As the objective function the authors use a weighted sum of four measures, namely the number of districts, the compactness of subdistricts, district similarity in subsequent periods and balance with respect to salesmen's profit. They propose an Adaptive Large Neighborhood Search and solve modified Solomon and Gehring & Homberger test instances with up to 400 customers and a maximum of three periods. Lei et al. (2016) describe a similar problem, in which customers are either deterministic or stochastic. Districts must be determined for each period of the planning horizon before the stochastic customers are revealed. All customers (deterministic and stochastic) of the same district have to be served on a single vehicle route from a central depot. The objectives are the same as in Lei et al. (2015), but instead of using a weighted sum as the objective function, the authors treat the problem as a true multi-objective optimization problem and solve it with a multi-objective evolutionary algorithm. Although the problems in Lei et al. (2015) and Lei et al. (2016) consider a multi-period planning horizon, they do not contain a scheduling component comparable to the MPSTDP-S. In Lei et al. (2015), the service days within each period must be decided, but, in contrast to the MPSTDP-S, each customer must be served exactly once per period and, hence, there are no restrictions on the temporal distribution of visits. In particular, Lei et al. (2015) do not consider week rhythms and weekday patterns, which are essential components of the MPSTDP-S. In Lei et al. (2016), there is no scheduling aspect at all since the customers that have to be served in a particular period are given by the concrete demand realization. Hence, a transformation of the MPSTDP-S to the problems studied in Lei et al. (2015) or Lei et al. (2016) is not possible.

The task of scheduling regular customer visits throughout a planning horizon arises also in some extensions of the vehicle routing problem and in multi-period scheduling problems. Since there exist different variants of regularity considered in these problems, we introduce a short classification. Figure 4 contains examples for the most important types of regularity. In the figure, we consider one exemplary customer and a planning horizon of four weeks and five days per week. The filled circles indicate the visiting days of the customer. Regularity type (1) means that the visiting weeks are periodically recurring, i.e., the number of weeks between consecutive visiting weeks is constant. In the example, the customer is visited every second week, beginning from the first week of the planning horizon. Regularity type (2) is similar to type (1), but refers to days instead of weeks. A customer is said to have regularity type (2) if the number of days between consecutive visits is constant. Regularity type (3) is a special case of type (1). Here, besides the periodicity with respect to visiting weeks, the weekdays on which the visits take place are the same in each visiting week. The customer in the example is visited biweekly on the second and fifth weekday. Finally, regularity type (4) is given if the number of days between consecutive visits is constant and the weekdays of the visits are identical throughout the planning horizon. Note that in the MPSTDP-S, regularity type (1) or (3) is considered, depending on the presence of weekday regularity requirements.

Scheduling and regularity aspects are considered in the Period Vehicle Routing Problem (PVRP) and the Inventory Routing Problem (IRP). In the classical VRP, customers must be assigned to vehicles and vehicle routes must be determined. The PVRP extends the classical VRP by a multi-period planning horizon in which customers must be visited several times. As an additional decision, the PVRP contains the selection of a feasible visit schedule for each customer. Regularity types (1) – (4) can be considered through an appropriate choice of valid visit schedules. For reviews on the PVRP, we refer the reader to Francis et al. (2008); Irnich et al. (2014). Recent papers on specific variants can be found in Archetti et al. (2015); Miranda et al. (2015); Rahimi-Vahed et al. (2015). We would like to stress one particular paper from the PVRP literature, namely the paper by Mourgaya and Vanderbeck (2007). The problem studied by Mourgaya and Vanderbeck is quite

Week	1					2					3					4				
Type of regularity / Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
(1) Periodic w.r.t. weeks	●			●								●	●							
(2) Periodic w.r.t. days		●		●		●		●		●		●		●		●		●		●
(3) Periodic w.r.t. weeks + weekday regularity		●			●							●			●					
(4) Periodic w.r.t. days + weekday regularity		●					●					●					●			

Figure 4: Examples for different types of regularity. A filled circle indicates that the customer is visited on that day.

similar to our problem as it is a tactical variant of the PVRP, in which customer visits are scheduled and assigned to vehicles in such a way that workload is balanced and compact clusters are achieved, whereas routing cost are not explicitly taken into account. But in contrast to our problem, their tactical model does not contain weeks as a separate time scale, i.e., they do not take into account the compactness of week clusters. Moreover, the planning horizon considered in their experiments consists only of up to six days, and it appears questionable if their column-generation-based heuristic can be applied to planning horizons of several months.

In the IRP, a supplier is responsible for replenishing the inventory of its costumers. To this end, products must be delivered to the customers on vehicle routes starting and ending at the supplier. Besides the routing decision, the decisions in the IRP include the timing and the quantities of the deliveries. Regularity type (2) can be observed, e.g., in the cyclic IRP studied by Raa and Aghezzaf (2009). A less restrictive approach is described by Coelho et al. (2012), who consider (among other consistency features) the possibility of specifying a minimum and maximum time interval between consecutive visits of the same customer, which results in regularity type (2) if the minimum and maximum time interval are set to the same value. Extensive reviews on the IRP can be found in Bertazzi et al. (2008); Coelho et al. (2014). Recent papers on specific variants are provided by Chitsaz et al. (2016); Dong and Turnquist (2015); Ekici et al. (2015); Li et al. (2016).

The main difference to our problem is that both the PVRP and the IRP explicitly aim at minimizing routing costs. In our problem, however, we aim at geographical compactness.

Another class of problems related to the MPSTDP-S are multi-period scheduling problems in which tasks have to be scheduled according to strict, predefined rhythms. In these problems, the time period between consecutive executions of a task is constant, corresponding to regularity type (2) with the only difference that time is not necessarily discretized into days. Applications of this kind of multi-period scheduling problems can be found in maintenance scheduling (e.g., Wei and Liu, 1983), processor scheduling (e.g., Korst et al., 1991), and logistics (e.g., Campbell and Hardin, 2005; Delgado et al., 2005; Kazan et al., 2012). However, these problems have in common that geographical aspects are not taken into account, i.e., compactness is not considered a relevant planning criterion. For this reason, solution approaches for this class of problems cannot directly be applied to the MPSTDP-S.

In summary, the main differences between the MPSTDP-S and the related problems are the following: The presented multi-period territory design problems do not contain a scheduling aspect comparable to the MPSTPD-S. The objective in the PVRP and IRP is to optimize routing cost, whereas in our problem compact week and day clusters are desired. Multi-period scheduling problems lack the consideration of any geographical aspects.

4. Mathematical Formulation of the MPSTDP-S

In this section, we state the subproblem MPSTDP-S as a mixed integer linear program. To this end, we introduce the following additional notation.

Let $P = \bigcup_{b \in B} P_b$ denote the set of all week patterns and $Q = \bigcup_{b \in B} Q_b$ the set of all weekday patterns. Then, for each week pattern $p \in P$ the parameter ψ_p^w is 1 if the week pattern contains week $w \in W$, and 0 otherwise. Analogously, ω_q^d states whether weekday pattern $q \in Q$ contains day $d \in D$. Due to the rigid week rhythms, it is easy to transform the service times t_{bj}, $j \in \{1, ..., \frac{|W|}{r_b} n_b\}$ into parameters t_b^w, which state the time for serving customer $b \in B$ in week $w \in W$, and parameters t_{bq}^d, which denote the time required for the service of customer $b \in B$ on day $d \in D$ if weekday pattern $q \in Q_b$ is selected. The average weekly and daily service times are denoted by $\mu^{week} = \frac{T}{|W|}$ and $\mu^{day} = \frac{T}{|D|}$, respectively, with $T = \sum\limits_{b \in B, j \in \{1, ..., \frac{|W|}{r_b} n_b\}} t_{bj}$ being the total service time over all customers. Parameters τ^{week} and τ^{day} define the maximal allowable percentage that the actual service times may deviate from the average weekly and daily service times, respectively. The week of day $d \in D$ is given by $\phi(d) \in W$. The distance from customer i to customer b is given by c_{ib}, $i, b \in B$.

We introduce the following decision variables.

$$g_{bp} = \begin{cases} 1 & \text{if week pattern } p \in P_b \text{ is assigned to customer } b \in B \\ 0 & \text{otherwise} \end{cases}$$

$$h_{bq}^w = \begin{cases} 1 & \text{if weekday pattern } q \in Q_b \text{ is assigned to customer } b \in B \text{ in week } w \in W \\ 0 & \text{otherwise} \end{cases}$$

These variables are sufficient to describe the temporal distribution of the visits, but they do not suffice to take into account the compactness criteria. As the compactness measure in our approach, we use the sum of the distances between the customers that are served on a particular day (week) and a customer that is selected as the cluster center for this day (week). Such a center-based compactness measure is quite common in literature (see, e.g., Fleischmann and Paraschis, 1988; Hess et al., 1965; Hojati, 1996; Salazar-Aguilar et al., 2011). There are also other ways to measure compactness, e.g., based on pairwise distances between customers. However, these measures are computationally intractable when incorporated into a MIP model and can, therefore, only be used for an a posteriori evaluation of solutions.

To integrate the compactness measure into the model, we introduce the following auxiliary variables.

$$u_{ib}^w = \begin{cases} 1 & \text{if customer } b \in B \text{ is assigned to week center } i \in B \text{ in week } w \in W \\ 0 & \text{otherwise} \end{cases}$$

$$v_{ib}^d = \begin{cases} 1 & \text{if customer } b \in B \text{ is assigned to day center } i \in B \text{ on day } d \in D \\ 0 & \text{otherwise} \end{cases}$$

$$x_b^w = \begin{cases} 1 & \text{if customer } b \in \text{ is the center in week } w \in W \\ 0 & \text{otherwise} \end{cases}$$

$$y_b^d = \begin{cases} 1 & \text{if customer } b \in B \text{ is the center on day } d \in D \\ 0 & \text{otherwise} \end{cases}$$

For a better overview, the notation used in the basic model of the MPSTDP-S is summarized in Table 1.

4.1. Basic Model

Using the introduced notation, the MPSTDP-S can be formulated as the following MIP, which we denote by $SCHEDULE_{MIP}$.

$$\lambda \sum_{b\in B} \sum_{i\in B} \sum_{w\in W} n_b c_{ib} u_{ib}^w + (1-\lambda) \sum_{b\in B} \sum_{i\in B} \sum_{d\in D} c_{ib} v_{ib}^d \rightarrow \min \tag{1}$$

s.t.

$$\sum_{p\in P_b} g_{bp} = 1 \qquad b \in B \tag{2}$$

$$\sum_{i\in B} u_{ib}^w = \sum_{p\in P_b} \psi_p^w g_{bp} \qquad b \in B, w \in W \tag{3}$$

$$u_{ib}^w \leq x_i^w \qquad b, i \in B, w \in W \tag{4}$$

$$\sum_{b\in B} x_b^w = 1 \qquad w \in W \tag{5}$$

$$\sum_{b\in B, p\in P_b} t_b^w \psi_p^w g_{bp} \geq (1-\tau^{week})\mu^{week} \qquad w \in W \tag{6}$$

$$\sum_{b\in B, p\in P_b} t_b^w \psi_p^w g_{bp} \leq (1+\tau^{week})\mu^{week} \qquad w \in W \tag{7}$$

$$\sum_{q\in Q_b} h_{bq}^w = \sum_{p\in P_b} \psi_p^w g_{bp} \qquad b \in B, w \in W \tag{8}$$

$$\sum_{i\in B} v_{ib}^d = \sum_{q\in Q_b} \omega_q^d h_{bq}^{\phi(d)} \qquad b \in B, d \in D \tag{9}$$

$$v_{ib}^d \leq y_i^d \qquad b, i \in B, d \in D \tag{10}$$

$$\sum_{b\in B} y_b^d = 1 \qquad d \in D \tag{11}$$

$$\sum_{b\in B, q\in Q_b} t_{bq}^d \omega_q^d h_{bq}^{\phi(d)} \geq (1-\tau^{day})\mu^{day} \qquad d \in D \tag{12}$$

$$\sum_{b\in B, q\in Q_b} t_{bq}^d \omega_q^d h_{bq}^{\phi(d)} \leq (1+\tau^{day})\mu^{day} \qquad d \in D \tag{13}$$

$$g_{bp} \in \{0,1\} \qquad b \in B, p \in P_b \tag{14}$$

$$h^w_{bq} \in \{0,1\} \qquad b \in B, q \in Q_b, w \in W \tag{15}$$

$$u^w_{ib} \geq 0 \qquad b, i \in B, w \in W \tag{16}$$

$$v^d_{ib} \geq 0 \qquad b, i \in B, d \in D \tag{17}$$

$$x^w_b \in \{0,1\} \qquad b \in B, w \in W \tag{18}$$

$$y^d_b \in \{0,1\} \qquad b \in B, d \in D \tag{19}$$

The Objective Function (1) aims at optimizing compactness. The first term represents the compactness of the week clusters, whereas the second term expresses the compactness of the day clusters. Parameter $\lambda \in [0,1]$ is used to weight between weekly and daily compactness. Constraints (2) guarantee that a valid week pattern is assigned to each customer. Constraints (3) and (4) ensure that a customer which is served in a particular week is assigned to a week center of the same week. Constraints (5) guarantee that exactly one week center per week is chosen. Balanced service times across the weeks are enforced by Constraints (6) and (7) by limiting the feasible deviation from the average weekly service time. Constraints (8) link the week pattern choice and weekday pattern choice for each customer. If the selected week pattern for a customer implies service in a particular week, a valid weekday pattern must be selected for this week. Otherwise, no weekday pattern may be selected. Constraints (9) – (13) are analogous to Constraints (3) – (7), but refer to decisions at day level instead of week level. Constraints (14) – (19) are the domain constraints. Note that Constraints (16) and (17) define continuous variables, but due to Constraints (3), (4), (9) and (10) these variables are implicitly binary.

Note that, due to the fact that the week patterns imply periodicity with respect to the visiting weeks of each customer, the basic model considers regularity type (1) for all customers.

4.2. Weekday Regularity

Recall that we defined weekday regularity as regularity with respect to the weekdays on which a particular customer is visited. We distinguish two variants, namely *strict weekday regularity* and *partial weekday regularity*. In the following, we describe the two variants and explain how model $SCHEDULE_{MIP}$ must be adapted in each case.

4.2.1. Strict Weekday Regularity

If strict weekday regularity is required for a particular customer, the customer must be visited according to the same weekday pattern in every visiting week. In other words, the weekdays on which the customer is visited must always be the same throughout the entire planning horizon. Hence, a customer with strict weekday regularity has regularity type (3).

Let $B_{strict} \subseteq B$ denote the set of customers that demand strict weekday regularity. Then, the following modifications of the model must be made. The first r_b weeks of the planning horizon contain exactly one

Table 1: Summary of the notation for the basic model of the MPSTDP-S

Index sets	
B	Customers
W	Weeks in the planning horizon
D	Days in the planning horizon
P	All week patterns
P_b	Valid week patterns for customer $b \in B$
Q	All weekday patterns
Q_b	Valid weekday patterns for customer $b \in B$
Parameters	
$c_{ib} \in \mathbb{R}^+$	Distance from customer $i \in B$ to customer $b \in B$
$n_b \in \mathbb{N}^+$	Number of visits of customer $b \in B$ per visiting week
$t_b^w \in \mathbb{R}^+$	Time for serving customer $b \in B$ in week $w \in W$
$t_{bq}^d \in \mathbb{R}^+$	Time for serving customer $b \in B$ on day $d \in D$ if weekday pattern $q \in Q_b$ is selected
$\phi(d) \in W$	Week of day $d \in D$
$\psi_p^w \in \{0,1\}$	Indicates whether week pattern $p \in P$ contains week $w \in W$ (1) or not (0)
$\omega_q^d \in \{0,1\}$	Indicates whether weekday pattern $q \in Q$ contains day $d \in D$ (1) or not (0)
$\mu^{week} \in \mathbb{R}^+$	Average weekly service time
$\mu^{day} \in \mathbb{R}^+$	Average daily service time
$\tau^{week} \in \mathbb{R}^+$	Maximum allowable deviation of the actual from the average weekly service time
$\tau^{day} \in \mathbb{R}^+$	Maximum allowable deviation of the actual from the average daily service time
$\lambda \in [0,1]$	Weight for weekly compactness
Variables	
$g_{bp} \in \{0,1\}$	Takes a value of 1 if and only if week pattern $p \in P_b$ is selected for customer $b \in B$
$h_{bq}^w \in \{0,1\}$	Takes a value of 1 if and only if weekday pattern $q \in Q_b$ is selected for customer $b \in B$ in week $w \in W$
$u_{ib}^w \in \{0,1\}$	Takes a value of 1 if and only if customer $b \in B$ is assigned to week center $i \in B$ in week $w \in W$
$v_{ib}^d \in \{0,1\}$	Takes a value of 1 if and only if customer $b \in B$ is assigned to day center $i \in B$ on day $d \in D$
$x_b^w \in \{0,1\}$	Takes a value of 1 if and only if customer $b \in B$ is selected as the center for week $w \in W$
$y_b^d \in \{0,1\}$	Takes a value of 1 if and only if customer $b \in B$ is selected as the center for day $d \in D$

week in which customer $b \in B_{strict}$ is visited. Since, in the presence of strict weekday regularity, the same weekday pattern must be selected in every visiting week, the weekday pattern which is selected for the first r_b weeks determines the weekday patterns for all remaining weeks of the planning horizon. Hence, for all customers that require strict weekday regularity, variables h_{bq}^w need to be introduced for the first r_b weeks only. For all $b \in B_{strict}$, Constraints (15) are therefore modified as follows.

$$h_{bq}^w \in \{0,1\} \qquad b \in B_{strict}, q \in Q_b, w \in W, w \le r_b \tag{15a}$$

Moreover, for all $b \in B_{strict}$, Constraints (8), which link the week pattern and weekday pattern decisions, also need to be introduced for the first r_b weeks only.

$$\sum_{q \in Q_b} h_{bq}^w = \sum_{p \in P_b} \psi_p^w g_{bp} \qquad b \in B_{strict}, w \in W, w \le r_b \tag{8a}$$

In Constraints (9), (12) and (13) all variables h_{bq}^w with $b \in B_{strict}$, $w > r_b$ must be replaced by the corresponding variables of the first r_b weeks. For this purpose, we define function $\bar{\phi}(b,d)$ for all $b \in B$, $d \in D$.

$$\bar{\phi}(b,d) = \begin{cases} \phi(d) & \text{if } b \notin B_{strict} \\ ((\phi(d) - 1) \bmod r_b) + 1 & \text{if } b \in B_{strict} \end{cases}$$

For all customers without strict weekday regularity, i.e., $b \notin B_{strict}$, $\bar{\phi}(b,d)$ returns the week that contains the given day $d \in D$. For all customers which require strict weekday regularity, i.e., $b \in B_{strict}$, $\bar{\phi}(b,d)$ returns the week within the first r_b weeks of the planning horizon that determines the weekday pattern for customer b in the week which contains day $d \in D$.

All occurrences of $\phi(d)$ in the original model are replaced by $\bar{\phi}(b,d)$ which yields the modified Constraints (9a), (12a) and (13a).

$$\sum_{i \in B} v_{ib}^d = \sum_{q \in Q_b} \omega_q^d h_{bq}^{\bar{\phi}(b,d)} \qquad b \in B, d \in D \tag{9a}$$

$$\sum_{\substack{b \in B,\\ q \in Q_b}} t_{bq}^d \omega_q^d h_{bq}^{\bar{\phi}(b,d)} \ge (1 - \tau^{day}) \mu^{day} \qquad d \in D \tag{12a}$$

$$\sum_{\substack{b \in B,\\ q \in Q_b}} t_{bq}^d \omega_q^d h_{bq}^{\bar{\phi}(b,d)} \le (1 + \tau^{day}) \mu^{day} \qquad d \in D \tag{13a}$$

4.2.2. Partial Weekday Regularity

Similarly to strict weekday regularity, partial weekday regularity also describes the requirement that a customer must be visited according to a regular weekday pattern. However, partial weekday regularity allows a predefined number of deviations from the regular weekday pattern and is, therefore, less restrictive than strict weekday regularity.

Let $B_{partial} \subseteq B$ denote the set of customers which require partial weekday regularity and $f_b \in \mathbb{N}^+$, $b \in B_{partial}$, the number of allowed deviations from the regular pattern for customer b. Then, for each customer $b \in B_{partial}$, additional variables and constraints need to be added to model $SCHEDULE_{MIP}$.

$$\sum_{q \in Q_b} h'_{bq} = 1 \qquad b \in B_{partial} \tag{20}$$

$$\sum_{w \in W} h_{bq}^w \ge h'_{bq} \left(\frac{|W|}{r_b} - f_b \right) \qquad b \in B_{partial}, q \in Q_b \tag{21}$$

$$h'_{bq} \in \{0,1\} \qquad b \in B_{partial}, q \in Q_b \tag{22}$$

Variables h'_{bq} defined in Constraints (22) describe whether weekday pattern $q \in Q_b$ is selected as the regular weekday pattern for customer $b \in B_{partial}$.

$$h'_{bq} = \begin{cases} 1 & \text{if weekday pattern } q \in Q_b \text{ is selected as the regular weekday pattern for customer} \\ & b \in B_{partial} \\ 0 & \text{otherwise} \end{cases}$$

Constraints (20) guarantee that for each customer $b \in B_{partial}$ exactly one regular weekday pattern is selected. $\frac{|W|}{r_b}$ is the number of weeks in which customer $b \in B_{partial}$ is visited throughout the planning horizon. Hence, Constraints (21) make sure that the selected weekday patterns deviate in at most f_b weeks from the selected regular weekday pattern.

4.3. Remarks on the Model

Using model $SCHEDULE_{MIP}$, we tried to compute optimal solutions for small test instances with 30 and 50 customers, four weeks and five days per week. Only three out of ten 30-customer instances could be solved to optimality within a time limit of one hour. The average optimality gap of the remaining seven 30-customer instances was 3.6%. Out of the ten 50-customer instances, none could be solved to optimality, even with a time limit of ten hours (the average optimality gap was 4.5%). Hence, it seems impossible to solve this model to optimality for realistic instance sizes, which typically comprise more than 100 customers and several months. This is mainly due to two reasons, namely the high symmetry of the model and the great number of variables. In the following, we describe our attempts to address these two issues.

Model $SCHEDULE_{MIP}$ contains variables to describe the selection of week patterns, g_{bp}, and variables to describe the selection of weekday patterns within weeks, h^w_{bq}. The weekday pattern variables contain more information than the week pattern variables. In fact, the values of the week pattern variables can be derived from the values of the weekday pattern variables. It is easily possible to formulate the MPSTDP-S without week pattern variables g_{bp} and, hence, reduce the number of variables in the model. But experiments showed that the performance of the model is better if it contains both weekday and week pattern variables. Therefore, we decided to use both groups of variables.

There is a lot of symmetry in model $SCHEDULE_{MIP}$, i.e., there exist many different feasible solutions that have the same objective function value. For example, consider the case where the week rhythm, r_b, is from the set $\{1, 2, 4\}$ and the number of visits per visiting week, n_b, is equal to one for all customers $b \in B$. Suppose that there are no weekday regularity requirements and no restrictions in terms of valid weekdays, i.e., the set of valid weekday patterns, Q_b, $b \in B$, contains a valid pattern for each weekday. Further, let the planning horizon consist of four weeks and five days per week. Let a given feasible solution consist of the four week clusters C^1, C^2, C^3 and C^4, which represent the customers that are scheduled for week one, two, three and four, respectively. Symmetric solutions can be determined by assigning the week clusters to different weeks. However, this rearrangement is subject to restrictions due to the customers' week rhythms. Customers with a week rhythm of one or four do not impose any restrictions on the rearrangement. But due

Table 2: Example: Rearrangements of week clusters yield symmetric solutions.

	Visit in week			
Symmetric solution no.	1	2	3	4
1	C^1	C^2	C^3	C^4
2	C^3	C^2	C^1	C^4
3	C^1	C^4	C^3	C^2
4	C^3	C^4	C^1	C^2
5	C^2	C^1	C^4	C^3
6	C^4	C^1	C^2	C^3
7	C^2	C^3	C^4	C^1
8	C^4	C^3	C^2	C^1

to the biweekly customers, week clusters C^1 and C^3 as well as week clusters C^2 and C^4 must not be assigned to subsequent weeks. Thus, eight symmetric solutions can be obtained by rearrangements of week clusters (assuming feasibility with respect to the balance constraints), see Table 2. Additionally, the model contains a lot of symmetry at the level of day clusters. Since there are no restrictions with respect to the weekdays on which customers are served, there are 5! different ways of assigning day clusters to weekdays within each week. In a four-week planning horizon, this results in $(5!)^4$ symmetric solutions due to rearrangements of day clusters. When the symmetry of week and day clusters is combined, $8 \cdot (5!)^4 - 1 = 1,658,879,999$ symmetric solutions can be determined to each feasible solution.

In order to deal with the high symmetry of the model, we tested instance-specific symmetry breaking constraints. The idea was to order the service times of the weeks and of the days within each week in such a way that many symmetric solutions become infeasible. However, we experienced a deterioration in the running times, presumably because the symmetry breaking constraints make it more difficult for the heuristics of the MIP solver to find new feasible solutions.

5. Location-Allocation Heuristic

Due to the high complexity of the problem, we propose a heuristic solution approach. Our approach – as many approaches in territory design – is based on the old idea of Hess et al. (1965) to decompose the problem into a location subproblem and an allocation subproblem (see Kalcsics et al., 2005, for an overview of papers using this idea). Therefore, we briefly describe the approach of Hess et al. in the following.

Hess et al. (1965) deal with a (single-period) political districting problem. In this problem, basic areas must be partitioned into electoral districts in such a way that the districts are compact, balanced with respect to population, and contiguous. In the location subproblem, they determine a subset of the basic areas which serve as district centers. For the first iteration of the algorithm, they use initial trial centers;

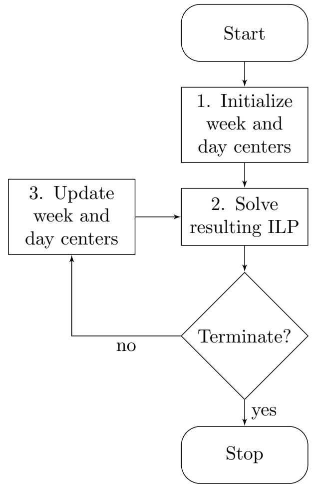


Figure 5: Location-allocation heuristic of Hess et al. (1965) adapted to the MPSTDP-S

for all subsequent iterations, they calculate the centers of gravity for each temporary district and use them as the new district centers. Then, in the allocation subproblem, they assign each basic area to exactly one district center. To this end, they solve a transportation problem and uniquely resolve all split assignments (a customer has a split assignment if he is assigned to more than one center). Location and allocation are repeated in an iterative manner until the solution process converges.

We adopt this decomposition approach for the MPSTDP-S. The general procedure of our adapted location-allocation heuristic is outlined in Figure 5. The algorithm starts with selecting an initial set of week and day centers (Step 1). By fixing the center decisions, we obtain an integer linear program (ILP) which is solved by a general-purpose MIP solver (Step 2). Then, the week and day centers are updated: For each week cluster and for each day cluster, the customer $b \in B$ which, when picked as the cluster center, leads to the smallest contribution to the Objective Function (1) is used as the new center (Step 3). Steps 2 and 3 are performed iteratively. The algorithm terminates if the current iteration has not produced an improved solution or if a user-defined maximum number of iterations, $iter_{max}$, has been performed.

To the best of our knowledge, our approach is the first that extends the work of Hess et al. (1965) to a multi-period setting. The major novelties of our location-allocation heuristic are the initialization procedure and the resulting ILP. In the following, we go into the details of these two components.

5.1. Selection of Initial Centers

The selection of good initial centers for the MPSTDP-S differs greatly from the single-period districting problem. In the single-period case, one wants to achieve compact, non-overlapping districts. Therefore, a reasonable strategy is to distribute the initial centers relatively evenly across the region under study, probably with a higher concentration in areas with high demand, i.e., in areas with a large number of customers or with a high level of activity. However, the strategy for the single-period districting problem is not applicable to the MPSTDP-S where customers are visited several times throughout the planning horizon. In the multi-period case, non-overlapping week or day clusters can, in general, not be achieved.

In the following, we develop a suitable initialization procedure for the MPSTDP-S based on the following observations.

1. At the level of individual customers, there is a weekly regularity due to week rhythms r_b, $b \in B$. These regularities can result in similarities at the level of week clusters, i.e., week clusters in different weeks may have a large number of customers in common. Such similarities can establish, at the earliest, after r_{min} weeks, with $r_{min} = min_{b \in B} r_b$ denoting the smallest week rhythm of all customers. To account for this, only r_{min} different initial week centers should be selected. If the number of weeks within the planning horizon, $|W|$, is greater than r_{min}, these week centers as well as their corresponding day centers should recur every r_{min} weeks.

2. The r_{min} different week centers should be evenly distributed over the entire region under study to facilitate the formation of compact week clusters, i.e., week clusters which span a relatively small geographical area.

3. The day centers of each week should obviously be close to their corresponding week center.

4. The day centers should, however, not (or at least not all) coincide with the corresponding week center, but rather be evenly distributed in the vicinity of the week center to promote the formation of compact day clusters.

5. The smaller the week rhythm r_b of a customer $b \in B$, the more likely it should be that the customer is selected as a week center or a day center. This favors the selection of customers $b \in B$ with $r_b = r_{min}$ and, therefore, increases the probability that the visits of these customers can be scheduled in accordance with their occurrence as centers.

We adapt the well-known initialization procedure of k-means++ (Arthur and Vassilvitskii, 2007), a popular seeding technique for cluster analysis, to take these observations into account. Let $c(b, J)$, $b \in B$, $J \subseteq B$ denote the minimum distance between customer b and any customer in set J. Then, given a set of candidate centers $I \subseteq B$ and the set of already selected centers $J \subseteq B$, the probabilistic function in Algorithm 1 is used to determine the next initial week or day center. Algorithm 1 is equivalent to the procedure used in k-means++ with the only difference that, in our adapted version, also the week rhythms are taken into account. Hence, in accordance with observations 2, 4 and 5, the probability that a candidate center is selected depends on its distance to the closest center already chosen and on its week rhythm. This means, the farther away from an already selected center and the smaller the week rhythm, the more likely it is that a customer is selected as the next initial week or day center.

The function in Algorithm 1 is used in Algorithms 2 and 3 to select the initial week and day centers, respectively. As in k-means++ (Arthur and Vassilvitskii, 2007), this is done in an iterative fashion, but we adapt the procedure of k-means++ in such a way that observations 1 and 3 are considered.

Algorithm 1 Function to pick the next initial week or day center based on k-means++ (Arthur and Vassilvitskii, 2007)

Input: Set of candidate centers $I \subseteq B$, set of already chosen centers $J \subseteq B$

Output: The next center $b \in I$

```
function NEXT_CENTER(I, J)
    if J = ∅ then
        return b ∈ I with probability (1/r_b) / Σ_{b'∈I} (1/r_{b'})
    else
        return b ∈ I with probability (c²(b,J)/r_b) / Σ_{b'∈I} (c²(b',J)/r_{b'})
    end if
end function
```

In the first while-loop of Algorithm 2, r_{min} different customers are selected as the week centers, $\gamma^w \in B$, $w \in W$ for the first r_{min} weeks of the planning horizon. The set of candidate centers consists of all customers, i.e., $I = B$. According to observation 1, the second while-loop makes sure that these centers repeat periodically every r_{min} weeks.

To select the initial day centers, $\gamma^d \in B$, $d \in D$, we proceed as illustrated in Algorithm 3. We subdivide the entire region into temporary week clusters by assigning each customer – independently of his week rhythm – to the closest week center, i.e., the temporary week cluster $\widetilde{C}^w$ is defined as $\widetilde{C}^w = \{b \in B : c_{\gamma^w b} < c_{\gamma^{w'} b}, w \neq w'\}$ for each week $w \in W$ with $w \leq r_{min}$. We use again the function in Algorithm 1 to determine suitable day centers, but we restrict the day center candidates to the customers within each temporary week cluster, i.e., $I = \widetilde{C}^w$. Through this, we make sure that the day centers of each week are close to the corresponding week center, as is required by observation 3. Analogously to the initialization of the week centers and according to observation 1, the day centers recur every r_{min} weeks.

An example of initial week and day centers is visualized in Figure 6. In this example, we assume that the planning horizon consists of $|W| = 8$ weeks and that the minimum week rhythm $r_{min} = 4$. Hence, the initial centers of week one correspond to the initial centers of week five, the initial centers of week two correspond to the initial centers of week six, and so on. The dashed lines indicate the borders of the temporary week clusters. The dark triangles represent the locations of the week centers and the light triangles the locations of the day centers within the respective weeks.

5.2. Integer Linear Program with Fixed Centers

When week and day center decisions are fixed, variables u^w_{ib}, v^d_{ib}, x^w_b and y^d_b (defined in Constraints (16) – (19)) can be removed from model $SCHEDULE_{MIP}$. The only remaining variables are the pattern variables g_{bp} and h^w_{bq} (Constraints (14) and (15)).

Note that the compactness criterion in the objective can now be expressed as a function of the pattern variables since the distances between customers and centers can be attached directly to the pattern variables.

Algorithm 2 Initialization of week centers

Input: Set of customers B

Output: Initial week centers γ^w, $w \in W$

```
procedure INIT_WEEK_CENTERS
    w ← 1
    J ← ∅
    while w ≤ r_min do
        γ^w ← NEXT_CENTER(B, J)
        J ← J ∪ {γ^w}
        w ← w + 1
    end while
    while w ≤ |W| do
        γ^w ← γ^(((w−1) mod r_min)+1)
        w ← w + 1
    end while
end procedure
```

Algorithm 3 Initialization of day centers

Input: Temporary week clusters $\widetilde{C}^w$, $w \in W$

Output: Initial day centers γ^d, $d \in D$

```
procedure INIT_DAY_CENTERS
    w ← 1
    while w ≤ r_min do
        J ← ∅
        for all days d in week w do
            γ^d ← NEXT_CENTER(C̃^w, J)
            J ← J ∪ {γ^d}
        end for
        w ← w + 1
    end while
    while w ≤ |W| do
        for all days d in week w do
            γ^d ← γ^(((d−1) mod (m r_min))+1)
        end for
        w ← w + 1
    end while
end procedure
```

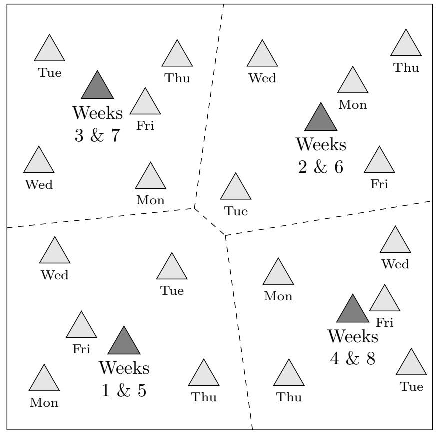


Figure 6: Example of initial week and day centers with $r_{min} = 4$ and $|W| = 8$

Denote again by $\gamma^w \in B$ the customer that represents the center of week $w \in W$, and by $\gamma^d \in B$ the customer that represents the center of day $d \in D$. Further, define $\bar{c}_{bp} = \sum_{w \in W} \psi_p^w n_b c_{\gamma^w b}$ and $\bar{c}_{bq}^w = \sum_{d \in D(w)} \omega_q^d c_{\gamma^d b}$ with $D(w)$ representing the days in week $w \in W$. Then, model $SCHEDULE_{MIP}$ reduces to the following integer linear program, which we denote by $ALLOC_{MIP}$.

$$\lambda \sum_{b \in B} \sum_{p \in P_b} \bar{c}_{bp} g_{bp} + (1 - \lambda) \sum_{b \in B} \sum_{q \in Q_b} \sum_{w \in W} \bar{c}_{bq}^w h_{bq}^w \rightarrow \min \tag{23}$$

s.t. (2), (6), (7), (8), (12), (13), (14) and (15).

If weekday regularity is required, this model is modified as described in Subsection 4.2.

6. Evaluation Measures

Recall that in the model $SCHEDULE_{MIP}$, we use a center-based compactness measure in the objective function because other compactness measures, e.g., measures based on pairwise distances, are computationally intractable. For the a posteriori evaluation of solutions, we are, however, not restricted to measures that are suitable for a MIP model. Hence, we use this section to do some groundwork for our extensive experiments in the next section by proposing appropriate measures to evaluate and compare solutions to the MPSTDP-S.

We introduce the following notation to represent solutions to the MPSTDP-S. Let C^{day} denote the set of day clusters and $C^d \in C^{day}$ denote the day cluster of day $d \in D$, i.e., $C^d = \{b \in B : b \text{ is served on day } d\}$. Analogously, denote by C^{week} the set of week clusters and by $C^w \in C^{week}$ the week cluster of week $w \in W$, i.e., $C^w = \{b \in B : b \text{ is served in week } w\}$. A solution to the MPSTDP-S is represented by the set of day and week clusters $C = \{C^{day}, C^{week}\}$. Note that the day clusters would be sufficient to fully describe a solution since the week clusters can be derived from the day clusters. Nevertheless, we use this redundant representation because this allows us to keep the formulation of the evaluation measures simple.

6.1. Compactness Measures

In the context of the MPSTDP-S, compactness refers to the geographical distribution of customers within the week and day clusters. Clusters with geographically concentrated customers are considered more compact than clusters that span a large geographical area. There are many ways to quantify the concept of compactness. We decided to use measures based on pairwise distances since this seems to be the most intuitive approach for our problem. More precisely, we measure the average distance between any two customers that belong to the same week or day cluster. The lower this distance, the higher is the geographical concentration of the customers in the cluster.

To evaluate the geographical compactness of the week clusters of solution C, we define the measure $WComp(C)$.

$$WComp(C) = \sum_{C^w \in C^{week}} \frac{\sum\limits_{b \in C^w} \sum\limits_{b' \in C^w,\, b \neq b'} c_{bb'}}{|C^w|(|C^w| - 1)} \tag{24}$$

Analogously, we define $DComp(C)$ to measure the geographical compactness of the day clusters of solution C.

$$DComp(C) = \sum_{C^d \in C^{day}} \frac{\sum\limits_{b \in C^d} \sum\limits_{b' \in C^d,\, b \neq b'} c_{bb'}}{|C^d|(|C^d| - 1)} \tag{25}$$

6.2. Travel Time Measures

The main motivation behind the compactness objective is the fact that the service providers have to travel to their customers and that geographically concentrated clusters are assumed to reduce the overall travel time. To account for this aspect, we propose additional measures based on route lengths. Please note that we assume that all daily routes start and end at the service provider's depot (e.g., the office or home), although, in practice, there can be overnight stays, meaning that the service provider does not return to the depot after all customers of the day have been served.

To evaluate a solution in terms of travel time, we solve a symmetric traveling salesman problem (TSP) for each day of the planning horizon and add up the daily travel times. The TSP for each day is defined on a complete graph. The nodes for day $d \in D$ correspond to the customers that are scheduled for that day, $C^d \in C^{day}$, plus the service provider's depot, E. Each pair of nodes is connected via edges and the edge cost corresponds to the travel time between the nodes. Let $\theta(N)$ be the travel time of an optimal solution (i.e., shortest travel time, optimality gap of max. 1%) to the TSP with nodes N. Then, the total travel time, $TT(C, E)$, of a solution C with depot E is calculated as the sum of travel times of the daily routes.

$$TT(C, E) = \sum_{C^d \in C^{day}} \theta(C^d \cup \{E\}) \tag{26}$$

The time needed to travel from the depot to the first customer of the daily route and from the last customer of the route back to the depot can only be reduced significantly if customers nearby the depot are assigned to the day cluster, even when other customers of the day cluster are far from the depot. In this case, the travel time from/to the depot is artificially decreased at the cost of a reduced cluster compactness. Apart from this undesirable case, daily compactness mainly effects the travel time *within* the day cluster, i.e., the travel time between customers. The travel time from/to the depot is more or less constant. Thus, it is interesting to have a measure which only considers the proportion of the total travel time that is related to trips between customers. For this purpose, we introduce the measure $TT_{IC}(C,E)$, which describes the total intra-cluster (IC) travel time of a solution C with depot E. Let $\eta(N,E)$ denote the travel time of an optimal solution to the TSP with nodes N minus the travel time associated with those edges of the solution that link the customers to the depot E. Then, measure $TT_{IC}(C,E)$ is defined as follows.

$$TT_{IC}(C,E) = \sum_{C^d \in C^{day}} \eta(C^d \cup \{E\}, E) \tag{27}$$

6.3. Balance Measures

Balance describes the requirement that the time needed to serve the customers should be evenly distributed throughout the planning horizon. This means that each day and each week should have roughly the same amount of service time. Perfect balance is achieved if the service time in each week is equal to the average weekly service time μ^{week}, and the service time on each day is equal to the average daily service time μ^{day}. As it is common in districting problems, we measure the maximum relative deviation from the average. We calculate the weekly balance, $WBal(C)$, and the daily balance, $DBal(C)$, of a solution C as follows:

$$WBal(C) = \max_{C^w \in C^{week}} \frac{|\chi(C^w) - \mu^{week}|}{\mu^{week}}, \tag{28}$$

$$DBal(C) = \max_{C^d \in C^{day}} \frac{|\psi(C^d) - \mu^{day}|}{\mu^{day}}, \tag{29}$$

where $\chi(C^w)$ is the service time that arises in week cluster $C^w \in C^{week}$, and $\psi(C^d)$ is the service time that arises in day cluster $C^d \in C^{day}$. The smaller the values of these measures, the more balanced we consider the solution.

7. Computational Experiments

We now present the results of extensive computational experiments. First, we report the results obtained from solving model $SCHEDULE_{MIP}$ on small test instances using the standard MIP solver Gurobi and derive some insights on the solution quality of our location-allocation heuristic. The main focus of this section is, however, on the evaluation of our location-allocation heuristic on test instances of realistic size. For this purpose, we develop an experimental design which covers a wide range of parameter values and problem characteristics. Since, for these realistic instance sizes, model $SCHEDULE_{MIP}$ cannot be solved by a standard MIP solver in a reasonable time, we benchmark our approach against the PTV xCluster Server (PTV, 2014), a commercial software product for scheduling customer visits. Additionally, we perform experiments to examine the impact of different types of weekday regularity on the travel time of the location-allocation solutions as well as on the running time behavior of the location-allocation heuristic, and we present a small extract of the solutions on a map.

7.1. Optimality Gap on Small Instances

As already mentioned in Subsection 4.3, we tried to compute optimal solutions to ten 50-customer test instances. The planning horizon for each instance consisted of four weeks and five days per week. We used the MIP solver Gurobi, warm started with the location-allocation solution, to solve model $SCHEDULE_{MIP}$. Gurobi could not find a (proven) optimal solution to any of the ten instances within a time limit of ten hours.[1] Hence, we do not know exactly how far the solutions of the location-allocation heuristic are from the optimal solutions. We can, however, compare the solutions of the location-allocation heuristic with the best incumbent and the best lower bound found by Gurobi for each test instance to obtain a range for the gap between the location-allocation solutions and the optimal solutions. We found out that the location-allocation solutions are, on average, 3.0% worse than the best incumbent found for each instance by Gurobi. On the other hand, the objective values of the location-allocation solutions are, on average, 8.0% higher than the best lower bound found by Gurobi. This means that the location-allocation approach produces high-quality solutions with an average optimality gap between 3.0% and 8.0%. The average runtime of the location-allocation approach was 4.6 seconds.

To provide a comparison with known optimal solutions, we briefly report in the following the results we obtain on the three 30-customer instances that could be solved optimally within one hour.[2] The optimality gaps for the location-allocation heuristic on these instances are 4.2%, 6.0%, and 7.3%. This means that high-quality solutions with an average optimality gap of 5.9% are found. The average running time per instance was 0.3 seconds.

7.2. Experimental Design

For the evaluation of the location-allocation heuristic we use 20 real-world instances provided by PTV. The data describe the planning task arising at a manufacturer of fast moving consumer goods whose sales force has to visit retailers, such as supermarkets and gas stations, on a regular basis. Each instance contains the service provider's depot and, on average, $|B| = 115$ customers. The customers' week rhythms, r_b, $b \in B$, are from the set $\{1, 2, 4, 8, 16\}$, which implies a planning horizon of $|W| = 16$ weeks. Each week consists of $m = 5$ days. All customers must be visited exactly once per visiting week, i.e., $n_b = 1$ for all customers $b \in B$. The weekdays on which visits may take place are not restricted, i.e., each weekday represents a valid weekday pattern $p \in P_b$ for all customers $b \in B$. The customers do not have weekday regularity requirements. Their service times (in minutes), t_{bj}, $b \in B$, $j \in \{1, ..., \frac{|W|}{r_b} n_b\}$, are from the set $\{22, 28, 34, 39, 42\}$, and each visit of a customer takes the same amount of time, i.e., $t_{bj} = t_{bk}$, for all $b \in B$, $j, k \in \{1, ..., \frac{|W|}{r_b} n_b\}$

In order to test our location-allocation heuristic under many diverse conditions, we generated additional test instances by modifying some parameters of the original real-world instances. The parameters we modified

[1] Gurobi version 6.0.2 was used for these tests. The tests were performed on a machine with an Intel Xeon E5-2650 v2 CPU with eight cores, running at 2.6 GHz, and 128 GB of RAM.

[2] Gurobi version 6.0.5, Intel Core i5-760, four cores at 2.8 GHz, 8 GB of RAM.

are the weekday regularity, the week rhythms, the number of visits per visiting week, and the service times (see Table 3 for a summary of the different parameter values that are covered by our test instances).

- *Weekday regularity:* First of all, we generated instances with strict weekday regularity for all customers as well as instances with partial weekday regularity for all customers. In the case of partial weekday regularity, we allowed one deviation from the regular weekday pattern, but we required that more than half of the visiting weeks of each customer must follow the regular weekday pattern. This means that all customers with at least three visiting weeks are allowed to deviate once from the regular weekday pattern, whereas all other customers are not allowed to deviate.

- *Week rhythms / Number of weeks:* With respect to the week rhythms, we generated instances in which all weekly customers of the original instances were changed to customers with a week rhythm of eight, and all biweekly customers of the original instances were changed to customers with a week rhythm of 16. This yields {4, 8, 16} as the set of week rhythms and a planning horizon of 16 weeks. Furthermore, we generated instances in which the week rhythms were randomly drawn from the set {3, 4, 6, 12, 16} with probabilities 15%, 20%, 30%, 20%, and 15%, respectively, resulting in a planning horizon of 48 weeks.

- *Number of visits per visiting week:* Concerning the number of visits per visiting week, we generated additional instances in which the number of visits per week were picked uniformly at random from the set {1, 2, 3}. Multiple visits per visiting week were, however, only eligible for weekly customers, since, from a practical point of view, it does not appear to make sense to serve non-weekly customers multiple times per visiting week. As in the original data, we assumed that there are no restrictions with respect to the combinations of weekdays on which visits may take place, i.e., the set of weekday patterns, P_b, $b \in B$, comprises all combinations of weekdays for which the number of contained weekdays equals the number of visits per visiting week.

- *Service times:* Finally, we generated additional instances by modifying the service times. For each visit of a customer, we picked a service time uniformly at random from the set {15, 20, ..., 55, 60}.

We choose a full factorial design, i.e., we consider all combinations of the above mentioned parameter values for all of the original 20 test instances. This yields, in total, 480 test instances. Using these instances, we perform computational experiments to compare the performance of the location-allocation heuristic with that of the PTV xCluster Server (PTV, 2014). Furthermore, we perform additional experiments to gain insights into the effect of weekday regularity on the travel time of the solutions as well as on the running time behavior of our algorithm.

7.3. Implementation Details and Parametrization

In the presence of partial weekday regularity requirements, the integer program $ALLOC_{MIP}$ must be modified as explained in Subsection 4.2. Additional variables and constraints must be added to $ALLOC_{MIP}$,

Table 3: Parameter values covered by the test instances

Parameter	Values
Weekday regularity	no regularity, partial regularity (1 deviation allowed), strict regularity
Week rhythms / Number of weeks in planning horizon	{1, 2, 4, 8, 16} / 16, {4, 8, 16} / 16, {3, 4, 6, 12, 16} / 48
Number of visits per visiting week	{1}, {1, 2, 3}[1]
Service times (minutes)	{22, 28, 34, 39, 42}, {15, 20, ..., 55, 60}

[1] Only in combination with week rhythms {1, 2, 4, 8, 16}

which makes it harder to find a feasible solution. To speed up the solution process, we first use the location-allocation algorithm to solve an auxiliary problem. In this auxiliary problem, all partial weekday regularity requirements are replaced by strict weekday regularity requirements, which, instead of introducing additional variables and constraints, leads to a reduction of the number of variables in model $ALLOC_{MIP}$. Note that the solution to the auxiliary problem is feasible for the original problem. Therefore, we use this solution to warm start the location-allocation algorithm on the original problem.

Both the location-allocation heuristic and the PTV xCluster Server (PTV, 2014) were run on a Windows 7 machine with 8 GB of RAM and an Intel Core i5-760 at a clock rate of 2.8 GHz. The location-allocation heuristic was coded in Java, and Gurobi 6.0.5 was used to solve model $ALLOC_{MIP}$. For all tests, the Gurobi MIP Gap parameter was set to 1%, which we consider sufficiently small for all practical applications. Moreover, the maximum time spent by Gurobi on solving the integer program in Step 2 of the algorithm was limited to 15 seconds. The maximum number of location-allocation iterations was set to $iter_{max} = 20$, which did not impose a restriction for the vast majority of the test instances in our experiments. In combination with the time limit of 15 seconds for the solution of the integer program, the maximum runtime of our heuristic is limited to five minutes per instance, which is according to our experiences with our industry partner PTV Group an acceptable computation time for human planners. If the objective function value did not improve by more than 0.1% compared to the previous iteration, the algorithm terminated early. The user parameter λ in Objective Function (23) was set to 0.33.

Depending on the focus of the experiments, we set the values of the balance tolerance parameters τ^{week} and τ^{day} differently. In Subsection 7.4, we compare the performance of the location-allocation heuristic and the PTV xCluster Server (PTV, 2014). For a fair comparison, we make sure that for all test instances the balance achieved with the location-allocation heuristic is at least as good as the balance of the PTV xCluster solution. To this end, we first solve each test instance with the PTV xCluster Server, and then use the values obtained for the weekly and daily service time balance as the values for the balance tolerance parameters of the location-allocation heuristic. As a consequence, all test instances in Subsection 7.4 are solved with different values for the balance tolerance. In Subsections 7.5 to 7.7, we focus on the impact of different types of weekday regularity on the travel time of the location-allocation solutions and on the running time behavior of the location-allocation heuristic. To guarantee the comparability of the results from this analysis, the same balance tolerance must be used for all instances. Therefore, we choose $\tau^{week} = 15\%$ as the weekly balance tolerance and $\tau^{day} = 30\%$ as the daily balance tolerance for all experiments in Subsections 7.5 to 7.7.

7.4. Comparison with PTV xCluster Server

Since the MPSTDP-S cannot be solved by a standard MIP solver for realistic instance sizes, we use the PTV xCluster Server (PTV, 2014) as the benchmark for the location-allocation heuristic. PTV xCluster Server uses a local search to determine a visit schedule that is valid with respect to the customers' visiting requirements. The optimization criteria of the local search are compactness and balance. At the beginning of the local search, the focus of the optimization is on improving compactness. During the course of the optimization, the focus shifts to the improvement of balance. Two types of moves are considered, namely the relocation of a customer to a different week or day cluster and the exchange of the week or day clusters of two customers. The algorithm terminates after a user-specified number of iterations or if no more improvements are found.

Remember that, for a better comparability of the location-allocation approach and the PTV xCluster Server (PTV, 2014), we set the balance tolerances, τ^{week} and τ^{day}, of the location-allocation heuristic to the actual service time balance of the xCluster solutions. Table 4 shows the average results of the two approaches with respect to compactness and travel time, grouped according to different types of weekday regularity. The first eight columns contain the average absolute values. $DComp$ and $WComp$ are measured in kilometers, TT and TT_{IC} are measured in hours. The last four columns show the relative deviation between the location-allocation solutions and the xCluster solutions with respect to the four measures. The relative deviation between the location-allocation solution $C_{LocAlloc}$ and the corresponding xCluster solution $C_{xCluster}$ on measure M is computed as

$$Dev(C_{LocAlloc}, C_{xCluster}, M) = \frac{M(C_{LocAlloc}) - M(C_{xCluster})}{M(C_{xCluster})}.$$

Hence, a negative deviation means that the location-allocation solution is better than the xCluster solution with respect to measure M. In the table, these deviations are averaged over all test instances of a row.

The results show that the location-allocation approach clearly outperforms the PTV xCluster Server (PTV, 2014) in all four compactness and travel time measures. With respect to measure $DComp$, the location-allocation solutions are, on average, 26.26% better than the xCluster solutions. Measure $WComp$ is improved by 13.47% compared to the xCluster solutions. The total travel time TT is reduced, on average, by 15.36 hours, the intra-cluster travel time TT_{IC} by 20.46 hours, which translates into relative improvements of 6.55% and 18.74%, respectively. It is noticeable that the reduction in the total travel time TT is smaller than the reduction of the intra-cluster travel time TT_{IC}. This means that the travel time between the depot and the day clusters increases compared to the xCluster solutions, but this increase is overcompensated by improvements of the intra-cluster travel time TT_{IC}. A possible explanation for this effect are outliers in the xCluster solutions, i.e., single customers that are relatively far from the other customers of a day cluster. Such

Table 4: Comparison between location-allocation approach and xCluster (PTV, 2014): Average compactness and travel time grouped according to the three types of weekday regularity

Weekday regularity	Location-allocation				PTV xCluster Server				Relative deviation between location-allocation and xCluster			
	$DComp$	$WComp$	TT	TT_{IC}	$DComp$	$WComp$	TT	TT_{IC}	$DComp$	$WComp$	TT	TT_{IC}
None	7.94	22.79	223.83	87.85	11.28	26.05	239.58	111.25	-30.88%	-13.42%	-7.18%	-22.42%
Partial	8.66	23.55	227.12	96.53	11.44	27.17	243.61	117.06	-25.55%	-13.74%	-6.72%	-18.14%
Strict	9.01	23.68	229.76	99.59	11.44	27.17	243.61	117.06	-22.34%	-13.25%	-5.74%	-15.66%
Average	**8.53**	**23.24**	**226.90**	**94.66**	**11.39**	**26.80**	**242.26**	**115.12**	**-26.26%**	**-13.47%**	**-6.55%**	**-18.74%**

Table 5: Comparison between location-allocation approach and xCluster (PTV, 2014): Relative compactness and travel time deviation grouped according to different sets of week rhythms and planning horizons

Week rhythms / Number of weeks in planning horizon	Relative deviation between location-allocation and xCluster			
	$DComp$	$WComp$	TT	TT_{IC}
{1, 2, 4, 8, 16} / 16	-20.19%	+0.49%	-5.70%	-13.47%
{4, 8, 16} / 16	-37.55%	-39.69%	-8.91%	-26.57%
{3, 4, 6, 12, 16} / 48	-27.10%	-15.17%	-5.89%	-21.46%
Average	**-26.26%**	**-13.47%**	**-6.55%**	**-18.74%**

outliers are, in some cases, produced by xCluster in an attempt to improve the balance of a solution. They can lead to a reduced travel time between the depot and the day cluster at the cost of intra-cluster compactness.

It can further be seen from Table 4 that, the higher the degree of freedom in terms of weekday regularity, the higher is the improvement of the location-allocation solutions over the xCluster solutions. For example, the average relative improvement on measure $DComp$ is 22.34% in the case of strict weekday regularity. When weekday regularity is relaxed to partial and none, the improvement increases to 25.55% and 30.88%, respectively. Similar effects can be observed for measures TT and TT_{IC}. Only on measure $WComp$ are the values almost the same for all three types of weekday regularity.

Table 5 provides a different view of the same results by grouping the relative deviation between the two approaches according to the three different sets of week rhythms and associated planning horizons. The location-allocation heuristic clearly beats xCluster (PTV, 2014) in all dimensions except one. When weekly customers are present, the $WComp$ values of the location-allocation approach and xCluster are nearly identical. This can be explained by the fact that the weekly customers force the service provider to travel almost across the whole service territory in every week, which leads to very similar solutions in terms of weekly compactness. In the cases without weekly customers, the location-allocation approach is able to produce solutions that have a significantly higher weekly compactness than the xCluster solutions.

Table 6: Comparison between location-allocation approach and xCluster (PTV, 2014): Average service time balance in percent

Weekday regularity	Location-allocation		PTV xCluster Server	
	$DBal$	$WBal$	$DBal$	$WBal$
None	45.69	11.31	46.03	12.02
Partial	21.40	7.10	21.46	7.48
Strict	21.36	7.01	21.46	7.48
Average	**29.48**	**8.47**	**29.65**	**8.99**

Table 7: Comparison between location-allocation approach and xCluster (PTV, 2014): Average and maximum running time in seconds

Weekday regularity	Location-allocation		PTV xCluster Server	
	Average	Max	Average	Max
None	14.54	103.07	14.80	73.40
Partial	41.57	156.26	7.23	33.60
Strict	25.94	109.27	6.96	32.41
Avg/Max	**27.35**	**156.26**	**9.66**	**73.40**

The average weekly and daily balance values, $WBal$ and $DBal$, are reported in Table 6. Remember that the balance tolerances τ^{week} and τ^{day} of the location-allocation approach were set to the actual balance values of the xCluster solutions. Consequently, the balance values of the two approaches are almost the same, with the location-allocation solutions having a slightly better balance.

Table 7 contains the average and maximum running times per instance in seconds. The location-allocation approach has significantly longer running times than the PTV xCluster Server (PTV, 2014). With an average of approximately 27 seconds, the location-allocation running times are almost three times as high as those of xCluster. However, one has to keep in mind that the MPSTDP-S is a tactical planning problem, which has to be solved only every few months. In such a tactical context, the location-allocation running times are completely acceptable. In fact, rather than having very short running times, solution quality is of utmost importance in practice since high-quality solutions can prevent the necessity of manual post-processing by a human planner.

7.5. The Cost of Weekday Regularity

In practice, many customers appreciate weekday regularity because it leads to a reduction in the time needed for coordination and to an increase in efficiency. However, enforcing partial or strict weekday regularity means that the solution space is restricted compared to the situation without weekday regularity. One would expect that such a restriction leads to a deterioration in the compactness and the travel time of the

Table 8: Cost of weekday regularity measured as the increase in travel time for the two types of service times relative to the case without weekday regularity

Weekday regularity	Service times		
	Original	Randomly picked	**Average**
Partial	+1.27%	+6.08%	**+3.68%**
Strict	+1.82%	+8.88%	**+5.35%**

solutions produced by the location-allocation approach. In this subsection, we investigate this "cost of weekday regularity". Concretely, we analyze the increase in travel time (measure TT) when weekday regularity is imposed relative to the situation without weekday regularity. Remember that we choose $\tau^{week} =$ 15% and $\tau^{day} = 30\%$ for all experiments in this subsection.

Table 8 contains the cost of weekday regularity for the two different types of service times considered in the test instances. On average over all 480 test instances, we observe a 3.68% increase in travel time when partial weekday regularity (max. one deviation from the regular weekday pattern) is enforced. In the case of strict weekday regularity, the total travel time is increased by 5.35%. A more detailed analysis shows that the cost of weekday regularity differs greatly depending on the values of the service times and week rhythms. The cost of weekday regularity is modest for instances with original service times: 1.27% in the case of partial weekday regularity and 1.82% in the case of strict weekday regularity. For the randomly generated service times, the cost of weekday regularity is much higher: It amounts to 6.08% and 8.88%, respectively. This result can be explained as follows. Remember that in the original real-world data all service times are from the set {22, 28, 34, 39, 42} and the same service time is incurred for each visit of the same customer. In our randomly generated test instances, the service time for each customer visit is randomly drawn from the set {15, 20, ..., 55, 60}, i.e., the service times may vary between different visits of the same customer. For example, a customer may require a 15-minute service on the first visit, a 60-minute service on the second visit and a 35-minute service on the third visit. Moreover, the range of the randomly drawn service times is more than twice as high as the range of the original service times. This means that there is more variability in the randomly drawn service times than in the original service times. When weekday regularity is imposed, the higher variability of the randomly generated instances leads to a greater increase in travel time.

Table 9 shows the cost of weekday regularity for the three types of week rhythms. Again, huge differences in the impact of weekday regularity can be observed. When week rhythms are from the sets {1, 2, 4, 8, 16} and {4, 8, 16}, the cost of weekday regularity is marginal (and even negative in one case). On the other hand, when the week rhythms are from the set {3, 4, 6, 12, 16}, weekday regularity leads to a significant increase in travel time of up to 18.51%. In the first two cases, all week rhythms are a power of two and, consequently, higher week rhythms are an integer multiple of smaller week rhythms. This facilitates the balancing of service times. The week rhythms in the third case do not have this beneficial property. Thus, the restrictions that

Table 9: Cost of weekday regularity measured as the increase in travel time for the three types of week rhythms relative to the case without weekday regularity

Weekday regularity	Week rhythms		
	{1, 2, 4, 8, 16}	{4, 8, 16}	{3, 4, 6, 12, 16}
Partial	+0.59%	-0.17%	+13.69%
Strict	+1.06%	+0.78%	+18.51%

go along with the introduction of weekday regularity cannot be compensated as easily as in the case of more favorable week rhythms.

In summary, we observed that enforcing weekday regularity leads to an increase in travel time. However, the extent of the increase is different under different circumstances. In our experiments, we identified the service times and the week rhythms as the major influencing factors on the cost of weekday regularity.

7.6. Running Time Analysis

Based on the experiments of Subsection 7.5, we now investigate the running time behavior of the location-allocation approach. The average and maximum running times are listed in Table 10, grouped according to different types of weekday regularity and week rhythms. The average running time over all test instances is roughly 28 seconds, the maximum running time is 280 seconds.

None and strict weekday regularity yield very similar running times of approximately 22 seconds on average and 140 seconds at the maximum. In contrast, partial weekday regularity results in significantly longer running times of 40 seconds on average and 280 seconds at the maximum. The reason for this is that we need to adopt a more involved procedure when partial weekday regularity requirements are present than in the other two cases. The additional variables and constraints that must be introduced to the model (see Subsection 4.2.2) make it hard for the MIP solver to find an initial feasible solution. Therefore, we perform two runs of the location-allocation heuristic consecutively (see Subsection 7.3). We first solve an auxiliary problem with strict weekday regularity and then take this solution to warm start the location-allocation heuristic for the problem with partial weekday regularity. This two-stage procedure is obviously more time-consuming than performing just a single run of the location-allocation heuristic as in the other two cases.

Regarding the week rhythms and the resulting planning horizons, one can see that the 48-week planning horizon results in considerably longer running times than the 16-week planning horizons (on average 77 seconds vs. 11 and 13 seconds, respectively). The running times for week rhythms {1, 2, 4, 8, 16} and {4, 8, 16}, both with a planning horizon of 16 weeks, are very similar.

Table 10: Running times of the location-allocation approach (values in seconds)

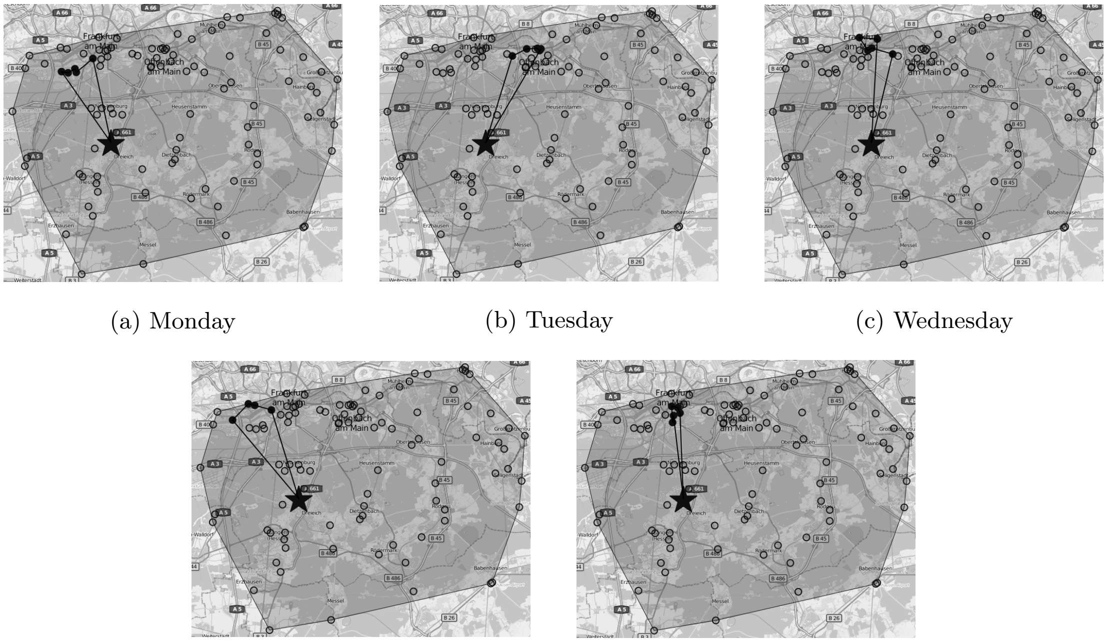
Weekday regularity	Week rhythms / Number of weeks in planning horizon							
	{1, 2, 4, 8, 16} / 16		{4, 8, 16} / 16		{3, 4, 6, 12, 16} / 48			
	Average	Max	Average	Max	Average	Max	**Avg/Max**	
None	11.17	50.95	10.36	57.80	54.66	141.10	**21.84**	**141.10**
Partial	13.18	174.09	15.90	47.67	118.00	280.04	**40.06**	**280.04**
Strict	8.94	97.56	13.14	35.50	57.99	139.96	**22.25**	**139.96**
Avg/Max	**11.09**	**174.09**	**13.13**	**57.80**	**76.88**	**280.04**	**28.05**	**280.04**

7.7. Visualization of Results

To give a visual impression of the solutions obtained with the location-allocation approach, we visualize the day clusters for the five working days of an exemplary week in Figure 7. The big star represents the service provider's depot, the circles represent the customers. A filled circle means that the customer must be served on that particular day, whereas an empty circle stands for a customer without a service request. The solid lines indicate the service provider's routes, which have been calculated a posteriori by solving a TSP for each day cluster. The darker area represented by the convex hull of the customers is the entire service territory, i.e., the region for which the service provider is responsible. The figure shows that the location-allocation approach produces geographically compact day clusters. Furthermore, all day clusters of the week are within a relatively small sub-area of the service territory, meaning that also a good weekly compactness could be achieved.

8. Conclusions and Future Research

In this paper, we introduced the multi-period service territory design problem. To the best of our knowledge, this problem has not been treated before in the literature, although its practical relevance is high. The MPSTDP combines two subproblems, namely a partitioning subproblem and a scheduling subproblem. Since the partitioning subproblem corresponds to the well-known (classical) territory design problem, we laid the emphasis of this paper on the scheduling subproblem. We formulated the scheduling subproblem as a mixed integer linear program. Due to the great number of variables and the high symmetry, it is – even on small instances – not possible to solve this formulation to optimality using a standard MIP solver. Therefore, we proposed a location-allocation heuristic. Extensive experiments on real-world instances and on instances derived from real-world data have shown that this heuristic produces high-quality solutions in reasonable running times. Our heuristic clearly outperforms the PTV xCluster Server (PTV, 2014) in terms of solution quality. Furthermore, we examined the cost of weekday regularity, i.e., the increase in travel time, when partial or strict weekday regularity is introduced. We found out that the cost of weekday regularity depends



(a) Monday (b) Tuesday (c) Wednesday

(d) Thursday (e) Friday

Figure 7: Day clusters and corresponding TSP routes for the five working days of an exemplary week (map data © OpenStreetMap contributors)

to a great extent on the characteristics of the test instances. The variability of the service times and the compatibility of the week rhythms have turned out to be the main influencing factors.

For the future, it is intended to integrate the location-allocation heuristic into the PTV xCluster Server (PTV, 2014). Beyond that, there are several possible extensions of the presented work. On the one hand, it would be interesting to investigate if at least small- or medium-sized instances of the MPSTDP-S can be solved to optimality by a more sophisticated exact solution method. On the other hand, there are several additional real-world planning requirements that could be integrated into the location-allocation heuristic. For example, it can be desirable in practice that the day clusters of consecutive days are close to each other because this strengthens the effect which motivates the weekly compactness criterion: To help the service provider catch up on visits of customers that have been missed on the scheduled day. Another interesting aspect is the integration of route length approximations. Although we are not interested in explicitly determining the service provider's daily routes, it could be beneficial to have an approximation for their (expected) duration. Such an approximation would allow to balance the total workload of the service provider, not just the service time. Another possible enhancement is the consideration of overnight stays of the service provider, which is a highly relevant aspect in many applications. Furthermore, it would be interesting to examine the suitability of a stochastic approach, which takes into account the uncertainty with respect to short-term customer requests and other unexpected situations that might occur in the daily business.

Acknowledgments

This work was partly supported by PTV Group and the German Academic Exchange Service (DAAD), Grant No. 57131828. This support is gratefully acknowledged. The authors thank PTV for the excellent cooperation and, in particular, for providing real-world problem data and the xCluster Server (PTV, 2014) as a benchmark for the location-allocation heuristic. Thanks are also due to the two anonymous referees for their valuable comments.

References

Archetti, C., Jabali, O., Speranza, M.G., 2015. Multi-period vehicle routing problem with due dates. Computers & Operations Research 61, 122–134. doi:10.1016/j.cor.2015.03.014.

Arthur, D., Vassilvitskii, S., 2007. k-means++: The advantages of careful seeding, in: Proceedings of 18th ACM-SIAM Symposium on Discrete Algorithms, SIAM. pp. 1027–1035.

Bertazzi, L., Savelsbergh, M., Speranza, M.G., 2008. Inventory routing, in: Golden, B.L., Raghavan, S., Wasil, E.A. (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, pp. 59–72. doi:10.1007/978-0-387-77778-8.

Blakeley, F., Argüello, B., Cao, B., Hall, W., Knolmajer, J., 2003. Optimizing periodic maintenance operations for Schindler Elevator Corporation. Interfaces 33, 67–79. doi:10.1287/inte.33.1.67.12722.

Campbell, A.M., Hardin, J.R., 2005. Vehicle minimization for periodic deliveries. European Journal of Operational Research 165, 668–684. doi:10.1016/j.ejor.2003.09.036.

Chitsaz, M., Divsalar, A., Vansteenwegen, P., 2016. A two-phase algorithm for the cyclic inventory routing problem. European Journal of Operational Research 254, 410–426. doi:10.1016/j.ejor.2016.03.056.

Coelho, L.C., Cordeau, J.F., Laporte, G., 2012. Consistency in multi-vehicle inventory-routing. Transportation Research Part C 24, 270–287. doi:10.1016/j.trc.2012.03.007.

Coelho, L.C., Cordeau, J.F., Laporte, G., 2014. Thirty years of inventory routing. Transportation Science 48, 1–19. doi:10.1287/trsc.2013.0472.

Delgado, C., Laguna, M., Pacheco, J., 2005. Minimizing labor requirements in a periodic vehicle loading problem. Computational Optimization and Applications 32, 299–320. doi:10.1007/s10589-005-4801-1.

Dong, Z., Turnquist, M.A., 2015. Combining service frequency and vehicle routing for managing supplier shipments. Transportation Research Part E: Logistics and Transportation Review 79, 231–243. doi:10.1016/j.tre.2015.05.002.

Ekici, A., Özener, O.Ö., Kuyzu, G., 2015. Cyclic delivery schedules for an inventory routing problem. Transportation Science 49, 817–829. doi:10.1287/trsc.2014.0538.

Fleischmann, B., Paraschis, J.N., 1988. Solving a large scale districting problem: A case report. Computers & Operations Research 15, 521–533. doi:10.1016/0305-0548(88)90048-2.

Francis, P.M., Smilowitz, K.R., Tzur, M., 2008. The period vehicle routing problem and its extensions, in: Golden, B.L., Raghavan, S., Wasil, E.A. (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, pp. 73–102. doi:10.1007/978-0-387-77778-8.

Groër, C., Golden, B., Wasil, E., 2009. The consistent vehicle routing problem. Manufacturing & Service Operations Management 11, 630–643. doi:10.1287/msom.1080.0243.

Hess, S.W., Weaver, J.B., Siegfeldt, H.J., Whelan, J.N., Zitlau, P.A., 1965. Nonpartisan political redistricting by computer. Operations Research 13, 998–1006. doi:10.1287/opre.13.6.998.

Hojati, M., 1996. Optimal political districting. Computers & Operations Research 23, 1147–1161. doi:10.1016/S0305-0548(96)00029-9.

Irnich, S., Schneider, M., Vigo, D., 2014. Four variants of the vehicle routing problem, in: Toth, P., Vigo, D. (Eds.), Vehicle Routing: Problems, Methods, and Applications. Siam. volume 18, pp. 241–271. doi:10.1137/1.9781611973594.

Kalcsics, J., 2015. Districting problems, in: Laporte, G., Nickel, S., Saldanha da Gama, F. (Eds.), Location Science. 1 ed.. Springer International Publishing, pp. 595–622. doi:10.1007/978-3-319-13111-5.

Kalcsics, J., Nickel, S., Schröder, M., 2005. Towards a unified territorial design approach — applications , algorithms and GIS integration. Top 13, 1–56. doi:10.1007/BF02578982.

Kazan, O., Dawande, M., Sriskandarajah, C., Stecke, K.E., 2012. Balancing perfectly periodic service schedules: An application from recycling and waste management. Naval Research Logistics 59, 160–171. doi:10.1002/nav.21479.

Korst, J., Aarts, E., Lenstra, J.K., Wessels, J., 1991. Periodic multiprocessor scheduling, in: PARLE'91 Parallel Architectures and Languages Europe, pp. 166–178. doi:10.1007/BFb0035103.

Lei, H., Laporte, G., Liu, Y., Zhang, T., 2015. Dynamic design of sales territories. Computers & Operations Research 56, 84–92. doi:10.1016/j.cor.2014.11.008.

Lei, H., Wang, R., Laporte, G., 2016. Solving a multi-objective dynamic stochastic districting and routing problem with a co-evolutionary algorithm. Computers & Operations Research 67, 12–24. doi:10.1016/j.cor.2015.09.002.

Li, M., Wang, Z., Chan, F.T., 2016. A robust inventory routing policy under inventory inaccuracy and replenishment lead-time. Transportation Research Part E: Logistics and Transportation Review 91, 290–305. doi:10.1016/j.tre.2016.05.001.

López-Pérez, J.F., Ríos-Mercado, R.Z., 2013. Embotelladoras ARCA uses operations research to improve territory design plans. Interfaces 43, 209–220. doi:10.1287/inte.1120.0675.

Miranda, P.A., Blazquez, C.A., Vergara, R., Weitzler, S., 2015. A novel methodology for designing a household waste collection system for insular zones. Transportation Research Part E: Logistics and Transportation Review 77, 227–247. doi:10.1016/j.tre.2015.02.019.

Mourgaya, M., Vanderbeck, F., 2007. Column generation based heuristic for tactical planning in multi-period vehicle routing. European Journal of Operational Research 183, 1028–1041. doi:10.1016/j.ejor.2006.02.030.

Polacek, M., Doerner, K.F., Hartl, R.F., Kiechle, G., Reimann, M., 2007. Scheduling periodic customer visits for a traveling salesperson. European Journal of Operational Research 179, 823–837. doi:10.1016/j.ejor.2005.03.056.

PTV, 2014. PTV xCluster Server version 1.18.1.3. URL: http://xserver.ptvgroup.com/en-uk/products/ptv-xserver/ptv-xcluster-ptv-xterritory/.

Raa, B., Aghezzaf, E.H., 2009. A practical solution approach for the cyclic inventory routing problem. European Journal of Operational Research 192, 429–441. doi:10.1016/j.ejor.2007.09.032.

Rahimi-Vahed, A., Crainic, T.G., Gendreau, M., Rei, W., 2015. Fleet-sizing for multi-depot and periodic vehicle routing problems using a modular heuristic algorithm. Computers & Operations Research 53, 9–23. doi:10.1016/j.cor.2014.07.004.

Salazar-Aguilar, M.A., Ríos-Mercado, R.Z., Cabrera-Ríos, M., 2011. New models for commercial territory design. Networks and Spatial Economics 11, 487–507. doi:10.1007/s11067-010-9151-6.

Wei, W.D., Liu, C.L., 1983. On a periodic maintenance problem. Operations Research Letters 2, 90–93. doi:10.1016/0167-6377(83)90044-5.

Zoltners, A.A., Sinha, P., 2005. The 2004 ISMS practice prize winner—sales territory design: Thirty years of modeling and implementation. Marketing Science 24, 313–331. doi:10.1287/mksc.1050.0133.